ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Departement of Mathematics

Semester Project

Spring 2000

Prediction of the Proteasome

http://dmawww.epfl.ch/~bloechli/projet2000

Ivo Blöchliger
Ivo.Bloechliger@epfl.ch

November 15, 2001
supervised by Alain Hertz

1 Introduction and overview of the project

1.1 Introduction to the problem

In each living cell unused proteins are cut by an enzyme called **Proteasome** into so called **peptides**. Peptides are small protein fragments of about 5 to 20 amino acids in length. These peptides are of a major interest in immunology since a cell presents them at its surface where they are recognized by the immunitary system.
The proteins and peptides are in fact strings of amino acids. Therefore the representation will be strings of the following 20 capital letters "A C D E F G H I K L M N P Q R S T V W Y" where each character represents an amino acid.
Being able to predict the peptides resulting from a given a protein would help to develop new ways of cancer treatment.
Due to the lack of data I didn't get very far with prediction, but I was able to determine which positions may have influence and which don't. Based on this analysis I developed a prediction method. But its efficiency has to be discussed with a biologist first.

1.2 Overview of the project

This project is divided into two parts. This is due to the fact that after about half of the time available for this project I got new and better but different data. Not only the structure of the data is quite different, but also the data results from completely different experiments. The first data set comes from an experiment in vitro on yeast proteins, the second comes from observations in human beings.

In the first part all data available I had was *one* yeast protein with a bunch of associated peptides originating from an experiment in vitro. For this kind of data I wrote a flexible computer program which was meant to predict the resulting peptides and not only the cleavage sites as it was done in preceding projects.

For the second part I was able to collect more voluminous, but quite different data. Frederic Levy [2] has kindly indicated a web page [3] out of which a data base of 290 peptides with their associated proteins has been built. Instead of a bunch of peptides for each protein only one peptide per protein was available. The program written for the first part was then obsolete. A statistical analysis of that new data allowed to show which positions (relative to the cutting position) of amino acids are important and which may be ignored in order to predict a cleavage site.

The work for this project was almost exclusively done on my private personal computer running **Linux**. Only free software was used. So this project should be portable on almost any well administrated computer system.

2 Part One

The available data [1] for this first part consists of the protein yeast Enolase 1 which was exposed in vitro to the enzyme Proteasome 20S. Although the data is very limited, I wrote a computer program based on this kind of data format in the hope to get more (much more) data later. Unfortunately this didn't happen, because either the data does not exist or people having this data are not willing to share it.

The work on this program was stopped when new data was available (part 2). The program is incomplete but I believe the chosen structure makes it quite flexible. For more precise information on the program please refer to the web pages in the `public_html/` directory of the project. (You may need to run "`make`" in the reports root directory in order to build the web pages.)

2.1 Basic ideas for the program

- No rules are given directly in the program. Instead all available data is loaded at the start of the program and then compared in various ways to the protein sequence to treat.

- The program will step through a given sequence of amino acids like the real Proteasome may do. It looks at the (lets say) twenty first positions after a given cleavage site. In this range one or more cleavage sites will be chosen and the program will treat the new sites as new starting points.

- The analyzing part of the program consists of a set of independent modules which evaluate the likelihood of a cut at a given position. The sum of the modules will then lead to a decision.

2.2 Conclusions and further work

The data produced by this program is not very promising for the moment. It may be a lot better if more data was available. Anyway I think its not very useful to search for general rules in such a small data set. One may continue this program if there is at least twenty times more data available.

It may be very useful to include results from part 2 into a predicting module. The main problem is that the experiments which produced the data sets in part 1 and part 2 are completely different. So they may not have that much things in common.

3 Part 2

Thanks to the help of Frederic Levy [2] I was able to build an important data base of proteins with one cleavage site each. In fact the data base consists of a collection of peptides for which the originating proteins were searched. These peptides were found on the surface of human cells. The peptides were not only

cut by the Proteasome, but may have been further selected by the process of the transport to the cells surface. An analysis of these peptides will of course take in account all this effects, and not only the Proteasome. This is the main reason why I think that the two data sets are not comparable and that prediction methods for one data set will not work for the other one. But there is no reason that the manner of analyzing the data sets should not be general.

3.1 Summary of the analysis for part 2

The distribution of the amino acids around the cleavage sites were analyzed and compared to the overall distribution of the amino acids. This allows to distinguish positions (relative to the cleavage site) with a potential influence on the cutting position from sites where the amino acids seem to appear randomly.

Based on the relative distribution of the amino acids around the cut a method was developed to predict the given cleavage site.

3.2 Data extraction

The data was extracted from:
"`http://134.2.96.221/Scripts/MHCServer.dll/CheckEp.htm`". A search for all strings matching **/HLA-[ABC].*/** was done. From the resulting page all peptides coming from "**Example for ligand**" were extracted and compared to the referenced protein sequence. The bad surprise was that only 359 out of 443 peptides matched in a unique manner a position in the protein. In most cases the peptides didn't match anything in the referenced protein, even if one typo was allowed... Out of the 359 peptides a bunch is located a the end of the protein and therefore of no interest for us, others appear twice. Finally there are 290 useful proteins left with one given cut each.

3.3 Statistical analysis

For the programs with which this analysis was performed, please refer to the computer program section 4.

3.3.1 What is the data?

The data consists of a set of 290 protein sequences. For each protein sequence **one** cutting position is given. For an example of the data base file please refer to 6.2.

3.3.2 Notation: What is a position?

A position always refers to a position in a protein relative to the given cutting position. The first amino acid after the cut has number 0, the second has number 1 and so on. The first amino acid before the cut has number -1, the second has number -2 and so on:

...	X	X	X	X	X	X	...
...	-3	-2	-1	0	1	2	...

3.3.3 Statistics on one position at a time

First the "global" distribution of the amino acids was computed. That means every amino acid in every protein of the data base was counted. A position will be considered having an influence on the cutting position if the amino acids appearing on that position do not follow the global distribution. K. Pearson's χ^2 test was used to determine the probability that a position follows the global distribution. This test furnishes a so called p-value. If it is less than 0.05 we can reject the hypothesis of randomness, else we cannot. This test is significant if for each amino acid the expectation under the hypothesis of randomness is 5 or more. With 290 proteins this is the case except for the amino acid 'W' where the expectation is only 3.

3.4 Results

3.4.1 Examples

The global distribution looks like figure 1. As an example the analysis for

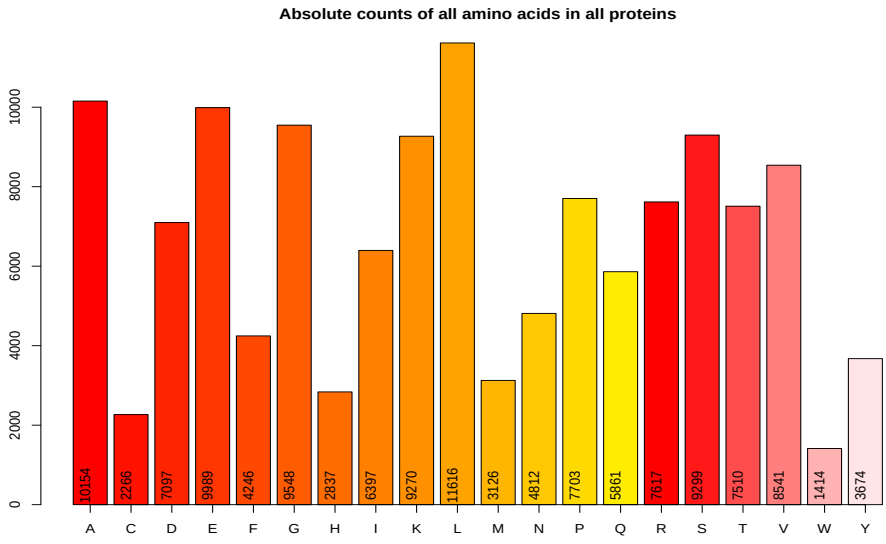

Figure 1: Global distribution of the amino acids in the data set

positions -1 and 1 will be shown as well:

- Figure 2 shows the absolute counts of the amino acids appearing at position -1 (immediately before the cut) . Figure 3 shows the normalized

counts that position. One clearly sees that the normalized distribution for position -1 does not follow a uniform distribution at all. We can conclude that position -1 is important to the cut. The statistical test confirms that with a p-value of 0.

- For position 1 (second position after the cut) things look quite different. If you compare the absolute counts (figure 4) to the global distribution (figure 1) you will see that they are quite similar. Indeed, the normalized distribution of the amino acids at position 1 (figure 5) looks a lot more like a uniform distribution. The χ^2-test confirms that with a p-value of 78%. The conclusion is that position 1 does not influence the cut which is a pretty surprising fact.

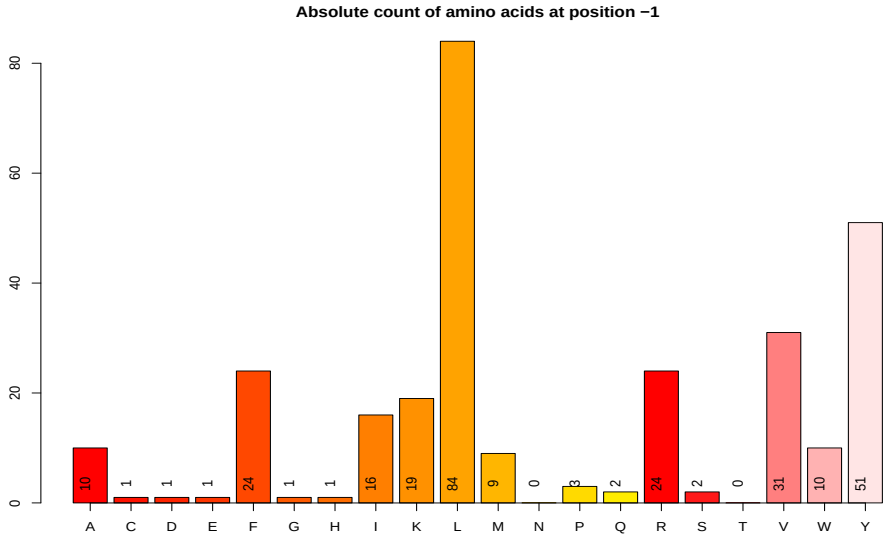

Figure 2: Absolute count of the amino acids appearing at the position right before the cut

3.5 All p-values

The p-values for position -20 to 10 are represented on figure 6. Please note that the cut is situated between position -1 and 0. Low values indicate a non randomness, values bigger than 0.1 indicate randomness. It is surprising to see that a window appears from position -10 to 0. This analysis shows that the important positions to determine the cut lie before the cut up to position -10 and that only one position after cut is relevant. The numerical values can be found in the data section 6.1.

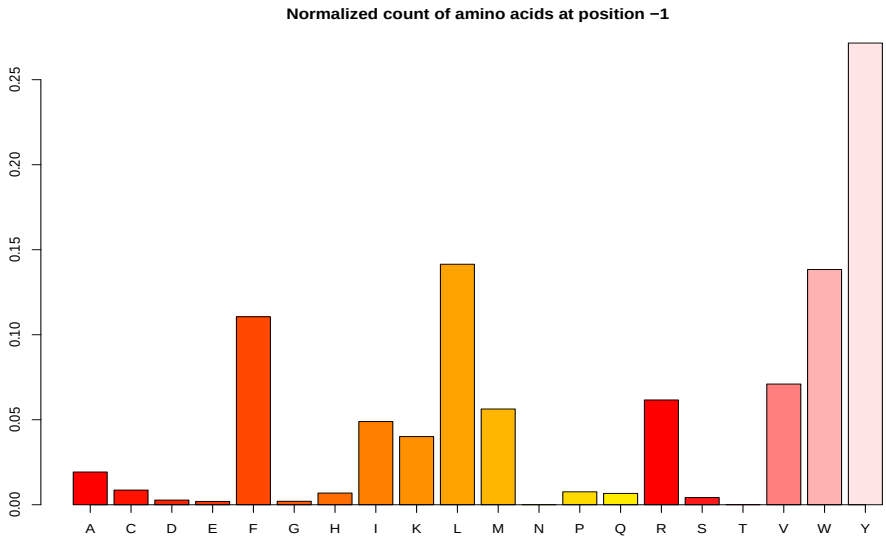

Figure 3: The distribution of the amino acids right before the cut normalized by the global distribution (figure 1)

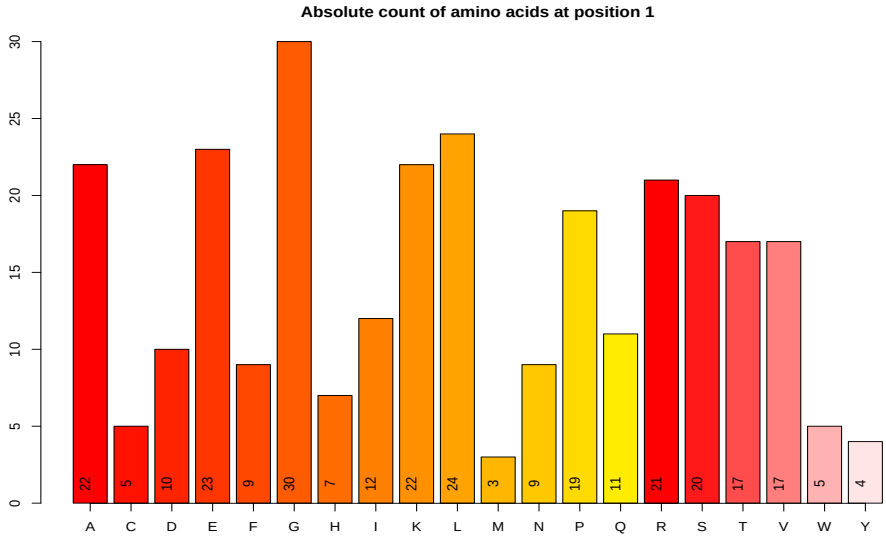

Figure 4: Absolute count of the amino acids appearing at the second position after the cut

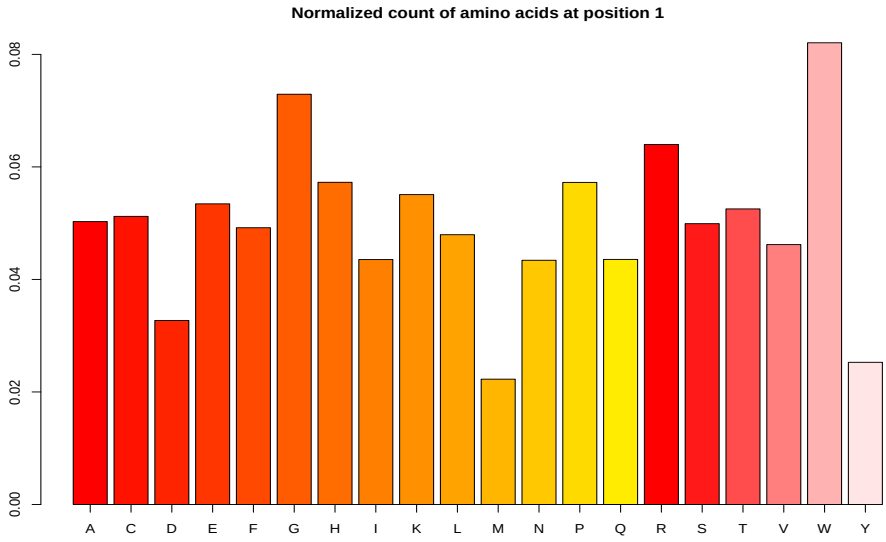

Figure 5: The distribution of the amino acids at the second position after the cut normalized by the global distribution (figure 1)

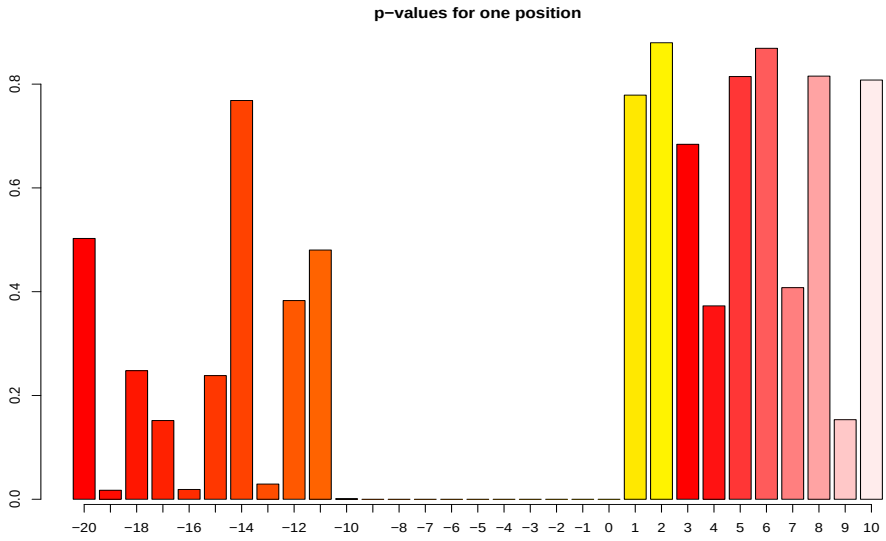

Figure 6: The p-value associated with each position. High values mean that the distribution at the corresponding position is random. Low values indicate an influence on the cutting position.

3.6 Predictions based on this analysis

The normalized distributions corresponding to specific positions can be used to predict the likelihood of a cut. To evaluate the likelihood of a cut for a position X one computes:

- a value y_i for each position i around X where y_i depends on the normalized count at the position i of the corresponding amino acid.
  - In fact : $y_i = 20 * f_{i,Z} + \frac{1}{10}$ where $f_{i,Z}$ is the normalized count at position i for the amino acid Z.
- Then the product of all y_i gives a number. The bigger it is, the more likely a cut may occur at this position X.
  - This likelihood function was the best out of a set of several functions. I tested $\sum_i y_i$ and $\prod_i y_i$ for different types of y_i like $y_i = 20 * f_{i,Z}$, $y_i = (20 * f_{i,Z})^2$ or $y_i = \sqrt{20 * f_{i,Z}}$. The addition of $\frac{1}{10}$ is useful for the case where a specific amino acid never appears at a certain position because this makes the whole product zero and therefore very unlikely. It is necessary to prevent that, since the data base is quite small. As the data base grows, this factor can be chosen closer to zero.

3.6.1 Results

It is quite difficult to test this method of predicting the cuts, since only *one* cut per protein is given. One step of the test looks like this:

- First the data base is divided randomly into 2 equally sized parts. Let's call them `random_base` and `random_test`.
- Based on the `random_base` data the normalized count data is built.
- Then for each protein in `random_test` to following is done:
  - In a window of 20 acids before and 20 acids after the real cut the likelihood for a cut is computed for every position.
  - The positions are sorted by their likelihood (biggest first).
  - The rank in the sorted list of the real cutting position is memorized. The lower the rank, the better the prediction. (Best is rank one, but rank five is still very good, since our window is 40 acids large)
- A histogram is updated with all ranks achieved.

This test is repeated a couple of times. For figure 7 the test was repeated 144 times, the window around the cut was from position -20 to 20. For the evaluation of the likelihood the normalized counts of positions -10 to 0 were used. 52% of the real cuts were ranked 3 or better, 64% were ranked 5 or better. If we allow to choose the 10 best positions out of 40 the chance is 80% to get also the real one.

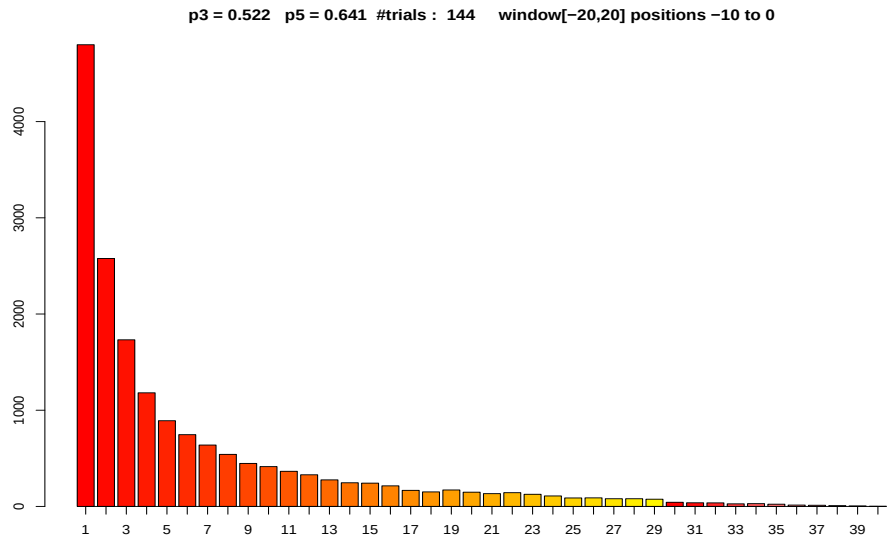

Figure 7: Histogram of achieved ranks by trying to predict half of the data based on the other half. (See 3.6 for more information)

3.6.2 Interpretation of the results

The quality of the prediction has to be discussed with a biologist first. The thing that is puzzling me is the fact that the distribution of the ranks seems to follow a very smooth curve. And more: the same curve appears for any window size and gets smoother as the number of tests grows. One can find more graphics for windows ranging from position -40 to 40 (`part2/one_out/50_out_80.ps`) and from positions -80 to 80 (`part2/one_out/50_out_160.ps`). It looks like it follows a stochastic law. Unfortunately I did not have the time to determine the corresponding law and its parameters. As a rule of thumb we can say that by choosing the best $\frac{n}{8}$ positions out of n the chance is 70% that the given cut has been chosen too. (70% were achieved while predicting *one* protein based on the 289 others. Look at `part2/one_out/one_out.ps`)

3.6.3 Is this method applicable to the Enolase from part 1?

Based on all data from part 2 I tried to predict the cutting positions for the enolase protein form part 1. Since the experiments are very different the result is not impressive at all. But even though the mass of the histogram in figure 8 lies to the left. The question is: do the ranks obtained for the Enolase follow the same distribution as the the ranks in figure 7?. If not we can conclude that the data for the two parts is inherently different and that there is little sense to apply results from part 2 to part 1. I strongly believe in that conclusion.

Unfortunately there is not enough data to perform a significant statistical test, but a χ^2 test furnished a p-value of 0 which supports the conclusion.

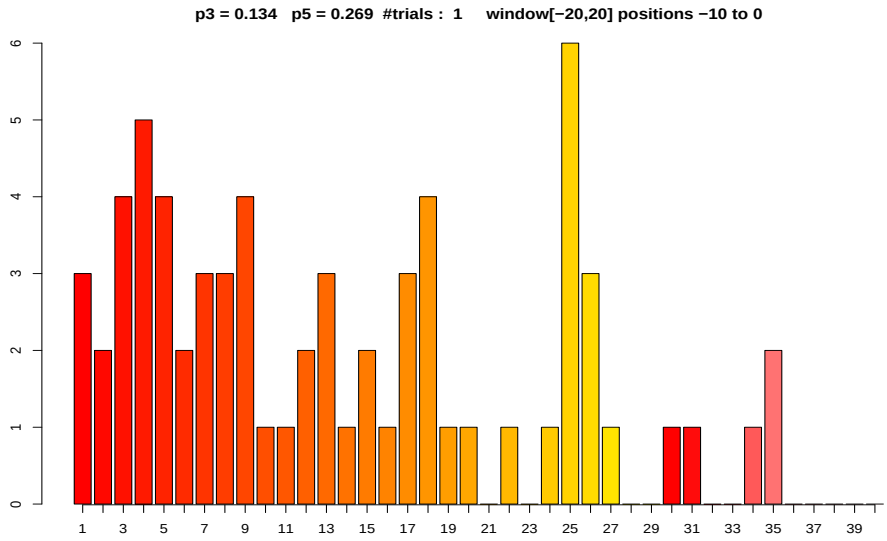

Figure 8: Histogram of achieved ranks trying to predict the cutting position of the enolase protein base on the all data from part 2

3.6.4 Prediction of two short protein fragments

Frederic Levy asked me to predict two small chains which were analyzed in vitro like the Enolase in part 1. I applied the prediction function of part 2 to them. The first problem is that my prediction function uses eleven positions (-10 to 0) in order to predict a cut. So the first 10 positions can not be predicted. I adjusted the function use smaller windows. The second problem is (as already mentioned above) that the two data sets may not have a lot in common. The results (see 6.3 and 6.4) seem quite random and of small significance. The given ranks correspond to the rank of the cut right **before** the considered amino acid. The double bars indicate the observed cuts in vitro.

4 Used and written computer programs

For the rest of this section I will assume that you have installed the source of this documentation somewhere on your account, since I will refer to subdirectories where all things are located. The following files and directories should be present:

```
ivo@gemini:~/2000projet/finalreport > ls -F
Makefile      data_base/    part2/        src/
README        html/         ps/           tex/
data/         listings/     public_html/
```

4.1 Needed Software

The work for this project was almost exclusively done on my home computer running Linux. **All software used is free software** in the sense of the GNU Public License GPL [5] (including the codes I wrote). This insures low costs for me, high stability of my computer system and portability on almost any well equipped computer system. The following computer program are needed:

- `g++` and `gcc` the GNU C++ and C compiler.
- `make` the GNU make utility.
- `R`[4] a GPL'ed version of "S-plus". R has the advantage to be free, to make much nicer graphics than S-plus and it knows how to make general χ^2 tests. (It won't work with S-plus, sorry)
- `perl` is needed for down-loading and building the data base for part 2, as well as for building the html documentation of the computer program from part 1.
- `lynx` a text based web-browser, needed to down-load the data from the Internet.
- `bash` the Bourne again shell, the standard Linux shell.

4.2 Codes I wrote

The source codes for part 1 of the project are located in the `src/` directory. There is a `Makefile` to build the program. For more information please refer to the html documentation in the `public_html/` directory. If it is empty either type "`make`" in the root directory of the report to build the complete report (including the LaTeXversion) or type "`make all`" in the `html/` directory to just build the html documentation in the `public_html/` directory. Then view `public_html/index.html` with *any* web browser.

For part 2 there are two types of programs: For down-loading and building the data base on one hand and for analyzing this data on the other hand. The programs for both categories were not meant to be user friendly or general, but just as tools to be used once. This is especially true for the programs that extracted the data from web pages.

In all directories containing source codes you will find also a `Makefile` to either build the programs and/or data or to clean up. Typing "`make clean`" erases all files that can be reconstructed by typing "`make all`"

4.2.1 Listings

If you want to print all the listings for this project go to the `listings/` directory and type "`make all`". (Of course, if you already typed "`make all`" in the reports root directory things are already in place.) This will build the files `all.ps` and `all.pdf` containing all codes. But before you print out all the 142 pages you might want to edit the file `all.tex` and comment out the parts you don't need. Type "`make clean`" to erase all files which are not needed. There will be 3 files left:

- `Makefile`
- `all.tex` : includes the files produced by
- `produce_listings.pl` : a perl script building the TEXsections out of source codes.

4.3 Building the data base

All data and programs needed to build the data base resides inside the `data_base` directory:

- `Makefile` to build or clean up the data base.
- `HLA-X.html` contains the whole web page with all the search results for HLA-A...to HLA-C...
- `goforit.pl` a Perl script for building the data base out of the web page `HLA-X.html`. The following files are produced:
  - `definitive/XXXX_YYYY` : For each peptide, one file is created where its name is formed of the peptide sequence followed by up to 10 characters corresponding to the amino acids following the peptide in the original protein. Such a file contains the peptide sequence, the whole protein sequence and the complete URL from where the protein sequence was down-loaded.
  - `emblfetch/XXXXXX` : The protein sequences are down-loaded from `http://www.ebi.ac.uk/cgi-bin/emblfetch?XXXXXX` where the 'X's are replaced by the protein code. For each protein the resulting page is saved in a file named after this code. This saves a lot of time and network traffic if one modifies the down-load script, since only unknown proteins are down-loaded. The others are reread from disk.
  - `mypeps/longpeps.txt` : This file contains the data that will be analysed. It contains two lines for each peptide. The first line contains the whole protein sequence, the second line contains an integer indicating the number of the last amino acid before the cut. (C programmers should rather consider that integer as the index of the first amino acid after the cut.) See 6.2 for an example.

4.4 The analyzer tools

All programs and data needed for the analyze are located in the `part2/` directory. There is also a `Makefile` to build or clean up. **Important:** to be able to use this tools you have to build the data base first. If you have typed "`make all`" in the projects root directory everything is already in place. Else run "`make all`" in the `data_base/` directory. Then run "`make all`" in the `part2/` directory.

4.4.1 `longanalyse.C` and `a.out`

This program performs the counting analysis of a data file containing the proteins with their cutting position. It invokes **R** to perform the χ^2 tests and to generate the histograms with the absolute and normalized count data and the p-values. Figures 2 to 6 were generated with this program. Typing "`make report`" in the `part2/` directory will regenerate this graphics.

The program takes the following options: (Note that there are no '-' before the options)

```
ivo@gemini:~/2000projet/finalreport/part2 > a.out -help

Usage:
--------
a.out [abs] [rel] [chi] [range n m] [dist d] [R path]
      [two] [count] [in DataFile]
abs : Makes histograms with absolute counts (ps/one_#_abs.ps)
rel : Makes histograms with normalised counts (ps/one_#_rel.ps)
chi : Makes graphics with the p-values (ps/chi_#.ps)
range n m : Scans from position n to m (-12..4)
            Positions:  ..., -2, -1 | 0, 1, ...
dist d : Looks for acids up to a distance of d. (4)
R path : Indicates the path to R. (R)
two : Make also an analysis on two positions
count : Make the counts and output to counts.dat
in Datafile : The datafile to be read. (data/longpeps.txt)
```

If `R` is in your `PATH` then there is no need to specify it. If not, or if your executable has another name than `R` you have to specify it. And again, it does not work with **S-plus** since it is not capable to perform a general χ^2 test.

The program is also able to consider two positions at a time (the option `dist` is used to specify the maximal distance of two acids to be considered). But this analysis is not significant at all. One would need at least 6000, better 10'000 peptides in order to get significant results.

4.4.2 `randomizer.C` and `randomizer`

This program splits randomly a data base file into two files of a given size, the files `random_base.txt` and `random_test.txt`. The first file is fed to the

`a.out` program which performs the counting analysis. The result of the counting analysis (file `counts.dat`) and the second file are fed to the `predict` program to perform the predictions of the data in `random_test.txt` based on the data in `counts.dat`. The data can be appended to an R-file `test.R` which will be read by the R-file `histogram.R`. In order to view the result do the following:

- Run R
- Type "`source("histogram.R")`" and the resulting histogram will appear.

To see the options for this program type "`randomizer -help`"
But rather than using `randomizer` use the script `runit` (see 4.4.4).

4.4.3 `predict.C` and `predict`

This program reads the file `counts.dat` (produced by `a.out`) and based on that it tries to predict a given file (normally the `random_test.txt`). The results (the ranks of the real cuts) are appended to a given R-file (normally `test.R`). Type "`predict -help`" for the usage.

4.4.4 `runit`

This is a bash script used to run the `randomizer` a lot of times in order to get good histograms. It was used to get figure 7. In case bash does not reside inside `/usr/bin` you may also run it with "`bash runit`".
Please edit this script if you wish to perform an analysis with different parameters. To view the results please refer to 4.4.2.

4.4.5 `predict2.C` and `p2`

This program is used to generate the ranks for all positions of a list of protein sequences. The tables from 6.3 and 6.3 are the direct output of this program. In order to use it, the file `counts.dat` has to be produced first. This can be done by typing "`a.out count range -10 0`" or any other desired range. There is a TEX file `test.tex` to view the output.

5 Conclusions

5.1 Part 1

Since there was not more data available I stopped the development of the program which worked on this kind of data. Based on the data I have seen, I believe that this kind of experiment in vitro is not well suited to predict the peptides which will finally be presented by the cell.

Trying to find general rules from one protein is, in my opinion, nonesens. Let's imagine you have book were no spaces are printed, and of course you don't know the language in which the book was written. Someone now gives you the

spaces for one small paragraph. Your task is now to find the place of the spaces for the rest of the book. It is not hard to find a small set of rules which will describe correctly the spaces in the given paragraph. But you will get pretty weird results for the rest of the book. Of course if you were given several pages one would be able to obtain much better results.

If at least 20 times more data is available it may be OK to continue the work on this program. It's a pity that I didn't get the new data of part 2 earlier. Not only there would be 3000 lines less of (for now) obsolete code, but I would have been able to work out a deeper analysis for part 2.

5.2 Part 2

The surprise of part two is clearly the distribution of the amino acids around the cut. The influent positions seem to lay in an asymmetric window from position -10 to 0 instead of an expected symmetric window of width about 6.

Also the prediction method seems pretty promising. I regret not having the time to analyze the distribution of the ranks (figure 7) since it seems to follow a stochastic law. The prediction may be enhanced a lot if not only the distribution of one position relative to the cut is analyzed, but the distribution of two positions simultaneously. But in order to do that we will need between 6000 and 10'000 peptides. I have no clue whether this data exists and if it will be accessible. But if this data is available some day, I would be pleased to continue or to assist this project. If the mentioned e-mail address should not be valid anymore, search for my full name on the Internet...

5.3 Thanks

I want to thank Alain Hertz for his help and availability. I also want to thank Frederic Levy; without him I would not have been able to build the data base for part 2 which made out of this project a success. Special thanks go to the thousands of people all over the world who contributed to all the free software I use every day.

Last but not least I want to thank my beloved girlfriend Thi Ngoc Tu HO for helping me with the statistical tests and reviewing this report.

Ivo Blöchliger, June 2000

6 The most important data files

6.1 The list of p-values corresponding to figure 6

```
Position -20          0.000975158           ------------
0.502631              ------------          Position 1
------------          Position -9           0.778819
Position -19          4.90056e-07           ------------
0.0173647             ------------          Position 2
------------          Position -8           0.87973
Position -18          9.99201e-16           ------------
0.247746              ------------          Position 3
------------          Position -7           0.684069
Position -17          1.41291e-09           ------------
0.151637              ------------          Position 4
------------          Position -6           0.372677
Position -16          1.89974e-08           ------------
0.0188883             ------------          Position 5
------------          Position -5           0.814571
Position -15          7.14801e-06           ------------
0.23827               ------------          Position 6
------------          Position -4           0.869038
Position -14          1.36339e-10           ------------
0.768479              ------------          Position 7
------------          Position -3           0.407776
Position -13          3.11561e-09           ------------
0.0292945             ------------          Position 8
------------          Position -2           0.815457
Position -12          1.60845e-10           ------------
0.382853              ------------          Position 9
------------          Position -1           0.153309
Position -11          0                     ------------
0.480288              ------------          Position 10
------------          Position 0            0.807886
Position -10          1.3392e-06            ------------
```

6.2 An example of the data base file for part 2

```
MAAADGDDSLYPIAVLIDELRNEDVQLRLNSIKKLSTIALALGVERTRSELLPFLTDTIY...
411
MSGYSSDRDRGRDRGFGAPRFGGSRAGPLSGKKFGNPGEKLVKKKWNLDELPKFEKNFYQ...
156
MHPFYTRAATMIGEIAAAVSFISKFLRTKGLTSERQLQTFSQSLQELLAEHYKHHWFPEK...
111
MSTNENANTPAARLHRFKNKGKDSTEMRRRRIEVNVELRKAKKDDQMLKRRNVSSFPDDA...
```

385
...

6.3 EFLWGPRALVETSYVK

The given ranks are associated with a cut **before** the corresponding amino acid. The double bars indicate cuts observed in vitro. So the smaller the ranks right after the double bars, the better.

6.3.1 Using positions -10 to 0

AA	E	F	L	W	G	P	R	A	L	V	E	T	S	Y	V	K
rank	*	*	*	*	*	*	*	*	*	*	1	4	5	6	2	3

6.3.2 Using positions -9 to 0

AA	E	F	L	W	G	P	R	A	L	V	E	T	S	Y	V	K
rank	*	*	*	*	*	*	*	*	*	3	2	6	7	5	4	1

6.3.3 Using positions -8 to 0

AA	E	F	L	W	G	P	R	A	L	V	E	T	S	Y	V	K
rank	*	*	*	*	*	*	*	*	6	2	3	5	7	8	1	4

6.3.4 Using positions -7 to 0

AA	E	F	L	W	G	P	R	A	L	V	E	T	S	Y	V	K
rank	*	*	*	*	*	*	*	9	5	1	3	7	6	8	4	2

6.3.5 Using positions -6 to 0

AA	E	F	L	W	G	P	R	A	L	V	E	T	S	Y	V	K
rank	*	*	*	*	*	*	6	10	8	1	4	5	7	9	3	2

6.3.6 Using positions -5 to 0

AA	E	F	L	W	G	P	R	A	L	V	E	T	S	Y	V	K
rank	*	*	*	*	*	11	6	10	5	3	4	8	7	9	1	2

6.3.7 Using positions -4 to 0

AA	E	F	L	W	G	P	R	A	L	V	E	T	S	Y	V	K
rank	*	*	*	*	3	12	9	11	5	4	6	7	8	10	2	1

6.3.8 Using positions -3 to 0

AA	E	F	L	W	G	P	R	A	L	V	E	T	S	Y	V	K
rank	*	*	*	7	2	13	10	9	6	4	5	11	8	12	1	3

6.4 DYLEETGSTAVPYGSFKHVDTRLQ

The given ranks are associated with a cut **before** the corresponding amino acid. The double bars indicate cuts observed in vitro. So the smaller the ranks right after the double bars, the better.

6.4.1 Using positions -10 to 0

AA	D	Y	L	E	E	T	G	S	T	A	V	P
rank	*	*	*	*	*	*	*	*	*	*	4	7

AA	Y	G	S	F	K	H	V	D	T	R	L	Q
rank	12	5	10	14	3	9	8	6	13	11	2	1

6.4.2 Using positions -9 to 0

AA	D	Y	L	E	E	T	G	S	T	A	V	P
rank	*	*	*	*	*	*	*	*	*	13	5	9

AA	Y	G	S	F	K	H	V	D	T	R	L	Q
rank	12	3	10	15	2	11	7	6	14	8	4	1

6.4.3 Using positions -8 to 0

AA	D	Y	L	E	E	T	G	S	T	A	V	P
rank	*	*	*	*	*	*	*	*	15	14	6	7

AA	Y	G	S	F	K	H	V	D	T	R	L	Q
rank	13	4	9	16	3	10	8	2	11	12	5	1

6.4.4 Using positions -7 to 0

AA	D	Y	L	E	E	T	G	S	T	A	V	P
rank	*	*	*	*	*	*	*	15	16	14	7	6

AA	Y	G	S	F	K	H	V	D	T	R	L	Q
rank	9	4	8	17	2	12	11	5	13	10	3	1

6.4.5 Using positions -6 to 0

AA	D	Y	L	E	E	T	G	S	T	A	V	P
rank	*	*	*	*	*	*	14	17	18	15	7	4

AA	Y	G	S	F	K	H	V	D	T	R	L	Q
rank	9	2	8	16	3	11	13	6	12	10	5	1

6.4.6 Using positions -5 to 0

AA	D	Y	L	E	E	T	G	S	T	A	V	P
rank	*	*	*	*	*	13	15	18	19	17	7	4

AA	Y	G	S	F	K	H	V	D	T	R	L	Q
rank	11	2	6	16	3	14	10	8	9	12	5	1

6.4.7 Using positions -4 to 0

AA	D	Y	L	E	E	T	G	S	T	A	V	P
rank	*	*	*	*	17	13	18	16	19	14	8	5

AA	Y	G	S	F	K	H	V	D	T	R	L	Q
rank	9	1	6	20	3	15	12	7	10	11	4	2

6.4.8 Using positions -3 to 0

AA	D	Y	L	E	E	T	G	S	T	A	V	P
rank	*	*	*	4	17	14	20	13	16	18	10	5

AA	Y	G	S	F	K	H	V	D	T	R	L	Q
rank	12	2	8	21	3	19	11	9	7	15	6	1

References

[1] Alexander K. Nussbaum et al. *Cleavage motifs of the yeast 20S proteasome β subunits deduced from digest of enolase 1*, Vol. 95 pp. 12504-12509, October 1998, Immunology

[2] Frederic Levy is leader of a research group at the *Ludwig Institude for Cancer Research in Epalinges, Switzerland*

[3] `http://134.2.96.221/Scripts/MHCServer.dll/CheckEp.htm`

[4] `http://www.ci.tuwien.ac.at/R/`

[5] `http://www.gnu.org`

Contents